

CAMUNDA

Camunda Meet-up

Presented by ilum:e informatik ag

September 2023



15. Camunda Meet-up

Harmonische Co-Existenz von Processengines und Legacy
Systemen mit eigenen Commerzbank-Framework

Camunda Chapter: Rhein-Main



ilum:e informatik ag
www.ilume.de

CAMUNDA

Platinum
Partner
CERTIFIED

Überblick: ilum:e informatik ag

Daten & Fakten

- Firmengründung im Jahr 2000 (Mainz)
- 150+ Expert*innen
- Beratung & IT Delivery in D-A-CH

Expertise & Kompetenzfelder

- **Smart Automation:** Automatisierung von Geschäftsprozessen & IT Abläufen
- **Customer Relationship Management:** Lösungen für Vertrieb, Marketing & Service
- **Softwareentwicklung:** Maßgeschneiderte Programmierleistungen



Die Community lebt vom Mitmachen!

Wer möchte beitragen?

- Erfahrungen aus Camunda Projekten aus Sicht des Kunden oder Implementierungspartner
- Deep-dives zu neuen Camunda-Features
- Neuigkeiten aus der Community
- Praktische Anwendung von Camunda bei Herausforderungen der Automatisierung und Digitalisierung
- Starke Use-Cases für Camunda

Beiträge gesucht!!!

Räumlichkeiten für ein
Meetup in Frankfurt
gesucht!!!

Nachlesen

Die Präsentation sowie ggfls. YouTube Links könnt ihr in den nächsten Tagen auf unserer Camunda Usergroup Seite gerne nochmal nachlesen/-sehen.

<https://ilume.de/sma/camunda-usergroup-rhein-main>





Co-Existenz von Processengines & Legacy Systemen

Ausgangssituation



- Großes Unternehmen – viele unterschiedliche Lösungen
- IBM Business Automation Workflow etabliert mit 8 Projekten
- Camunda vereinzelt
- Prozessmodellierung mit State Engine bei comdirect
- Inselhafte Prozesslösungen, keine übergreifende Transparenz in Sicht
- Schnelle vollautomatisierte aber auch „langsame“ Human Workflows
- Vielfalt an Interfaces
- Unterschiedliche Projekt-Lebenszyklen (langjährige Projekte)
- Wunsch nach Standard Prozessengine

Was brauchen wir wirklich?

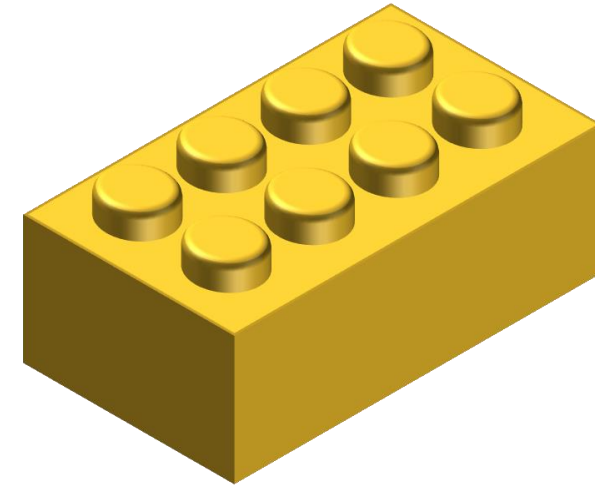


- Übersicht über Geschäftsprozesse (Modelle und laufende Prozesse)
„End2End“ vom Kundenauftrag bis zur Erledigung
- Migrationspfad / Integration bestehender Prozesse/Systeme (auch ohne Prozessengine)
- Technologische Freiheitsgrade für Cluster (Know How und Ressourcen)
- Verteilte Verantwortung (fachlich und organisatorisch / betrieblich)
- Kanalunabhängige Prozesse
- Fachliche APIs ohne Engine-spezifischen Details (~CoBa API Portal)

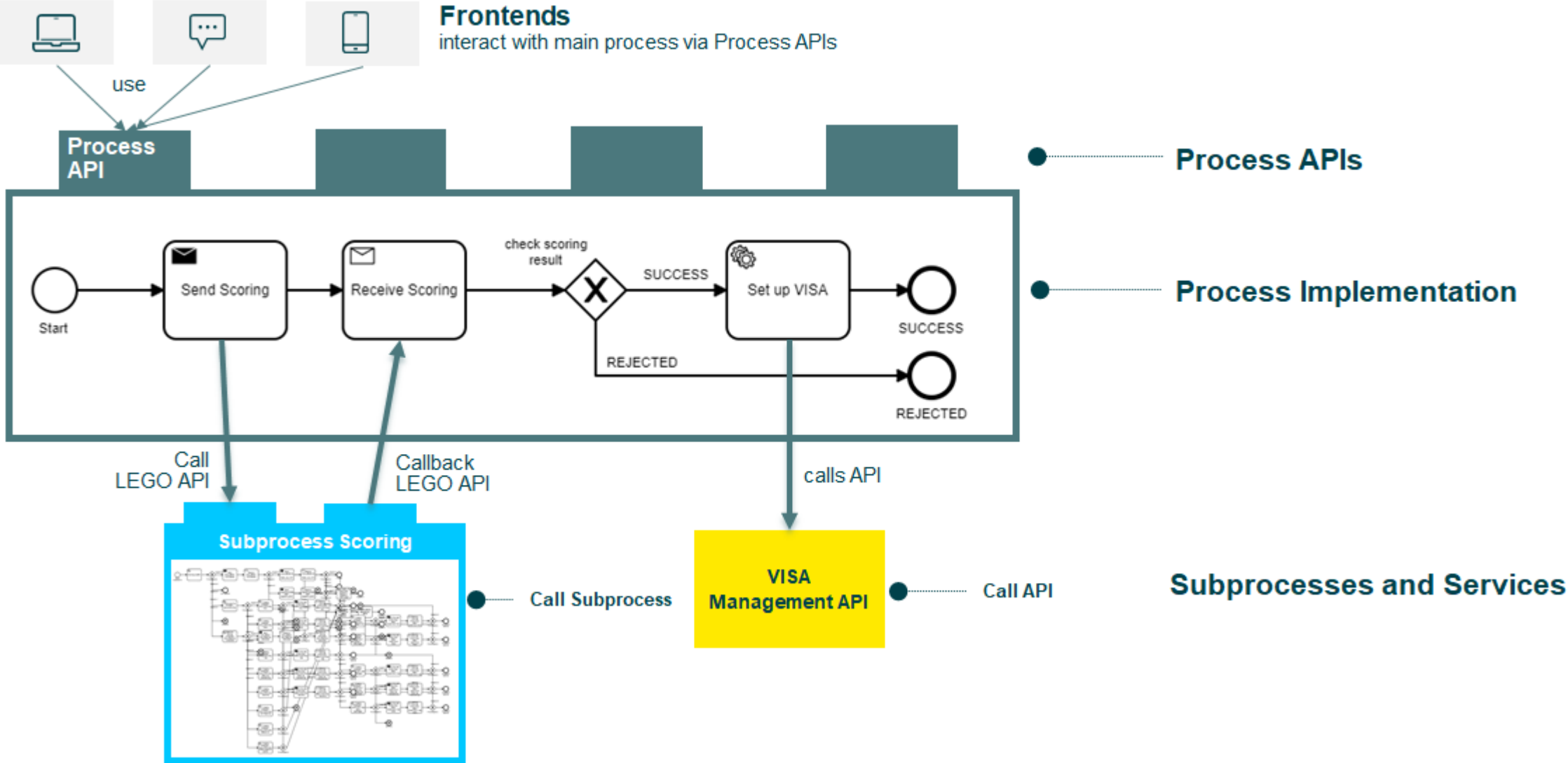
Lösungsansatz („LEGO Modell“)



- Prozesse per API kapseln
- Uniforme APIs mit standardisierten Metadaten
- Modularer Aufbau von End2End Prozessen aus Subprozessen
- Zentrales Repository für Prozessmodelle
- Zentrales Repository für Prozessstatus
- Verteilte Implementierung und Verantwortung
- Zentrale Dokumentation und zentrales Tracking
- So dezentral und autonom wie möglich, so zentralisiert und standardisiert wie nötig



LEGO Modell



Aspekte des LEGO Modells



- Modellierung und Strukturierung von E2E Prozessen
- Definition fachlicher Schnittstellen, Rollen und Verantwortlichkeiten

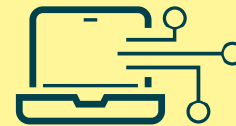


**Methodiken
und Vorgehen**

- Prozess Repository
- Status Repository und Tracking



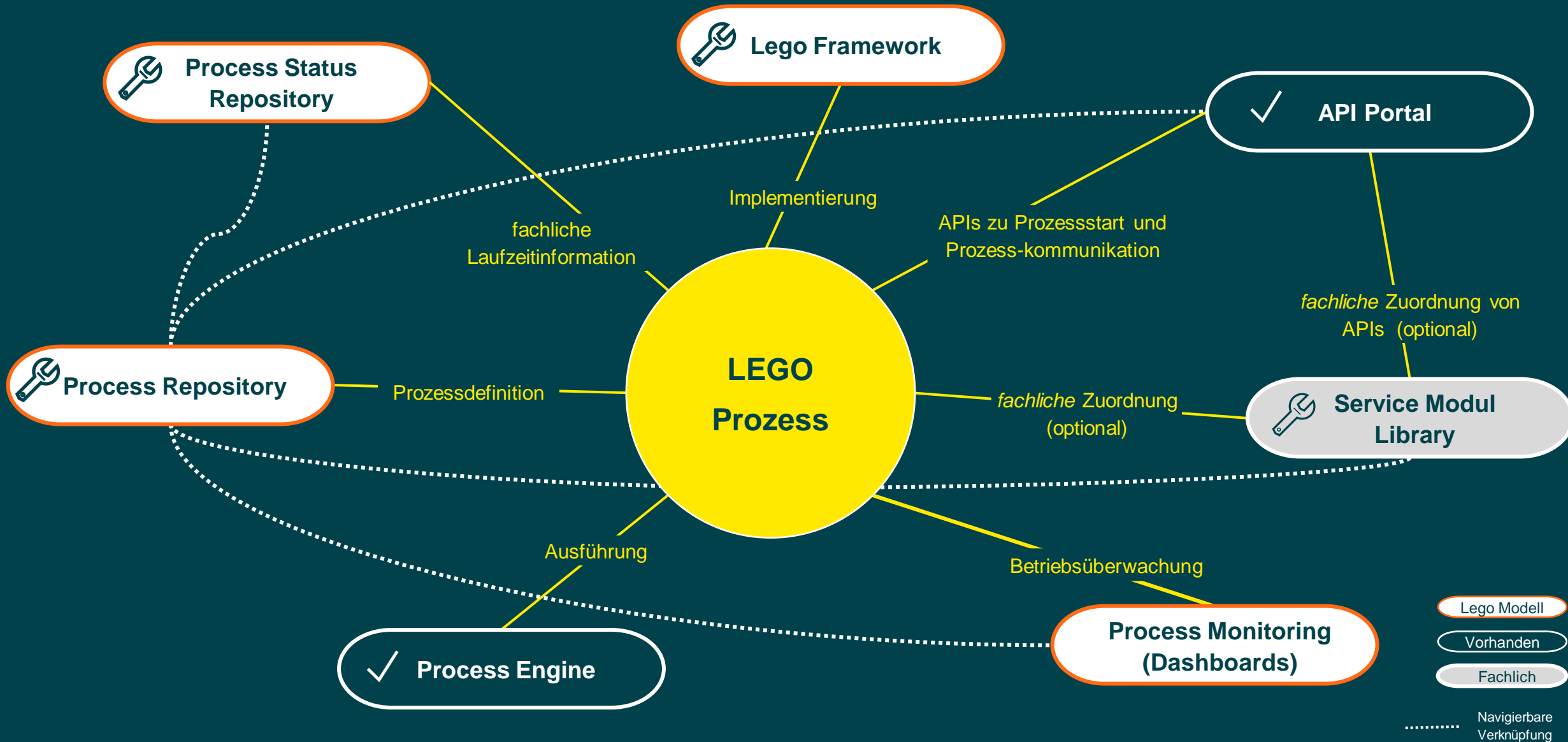
**Werkzeuge
und Hilfsmittel**



**Entwicklungs-
framework**

- Lego Framework
- Lego Prozess APIs
- BPM Engine Adapter (bspw. IBM BAW und Camunda)

Das Lego-Prozessuniversum

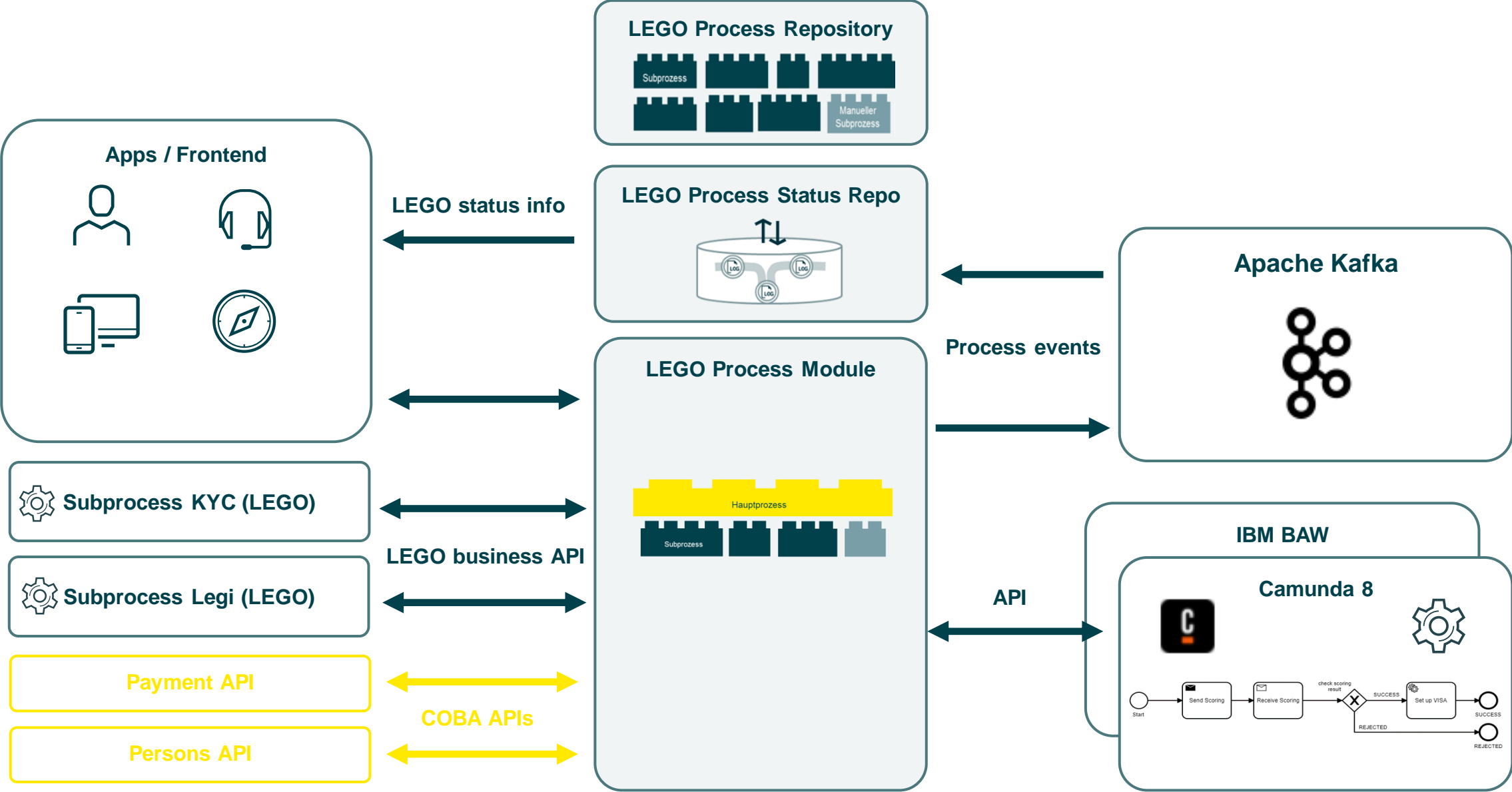


Implementierung



- Spring Boot Anwendungen für Prozesse
- Je Prozesstyp eine REST API
- API first, modularer Ansatz auch bei API
- Standardisierte Validierung für kanalunabhängige Verwendung
- Generierung von API und Schnittstellenimplementierung
- API zum Statustracking für Legacy Systeme
- Exporter für Statustracking und Callbacks
- Kapselung der Prozessengine

Module und Interaktionen





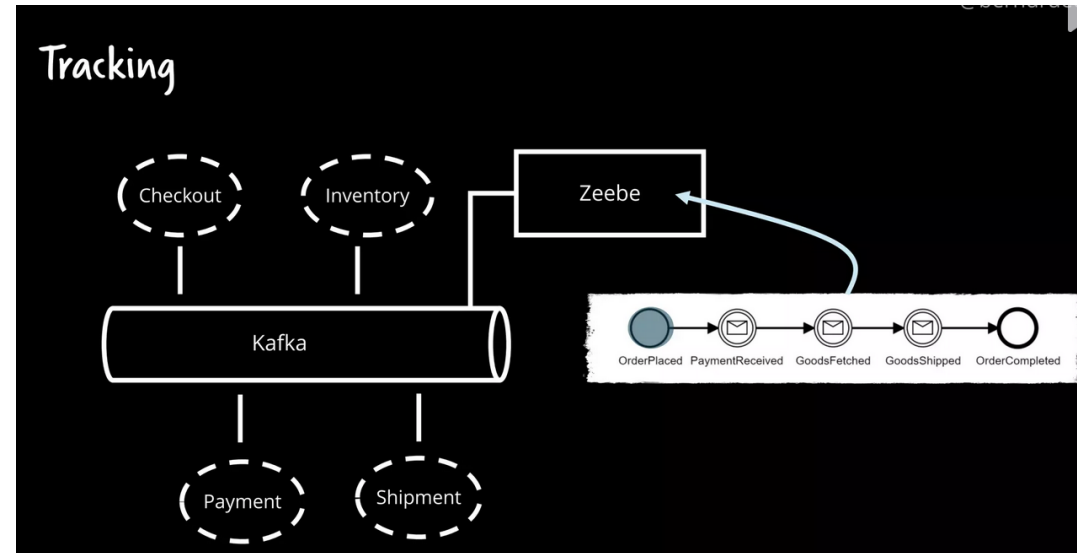
DEMO



Legacy Anbindung

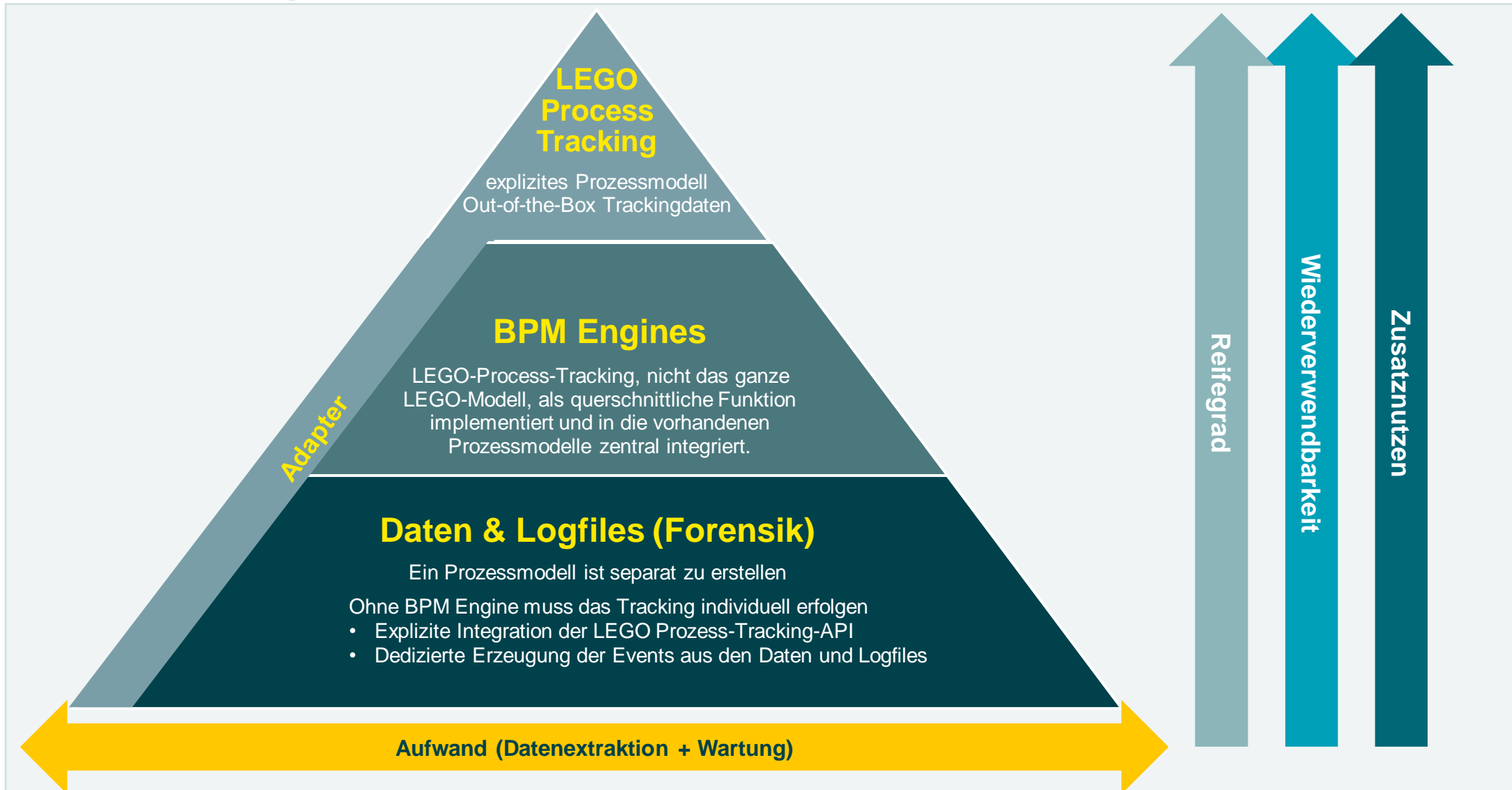


- Frei nach Bernd Ruecker 😊



- (https://es.slideshare.net/BerndRuecker/monitoring-and-orchestration-of-your-microservices-landscape-with-kafka-and-zeebe-meetup?next_slideshow=true)
- Entweder Tracking API nutzen oder Datenbanken/Logs minen
- Anbindung von Ticketsystem (ZeBRA) über standardkonforme APIs, Ticketbearbeitung integriert in Prozesstracking, viele Prozesse starten als manuelle Ticketbearbeitung und werden dann sukzessive automatisiert

Prozess-Status-Daten sammeln einfach ermöglichen – Anbindung Prozess-Status API (Adapter oder Lego-Modell)



Vorteile des Ansatzes



- Kein Vendor Lock In
- Kein vollständiger Migrationszwang bei Major Releases (z.B. Camunda 7->8)
- Querschnittsfunktionen über Framework abbildbar (Metriken, Logging, Security)
- Produktivitätsgewinn durch Standardisierung, Generierung und Templates
- Governance im Entwicklungsprozess berücksichtigt
- Langfristige Vorhaltung von Prozessdaten unabhängig von Prozessengines

Nachteile



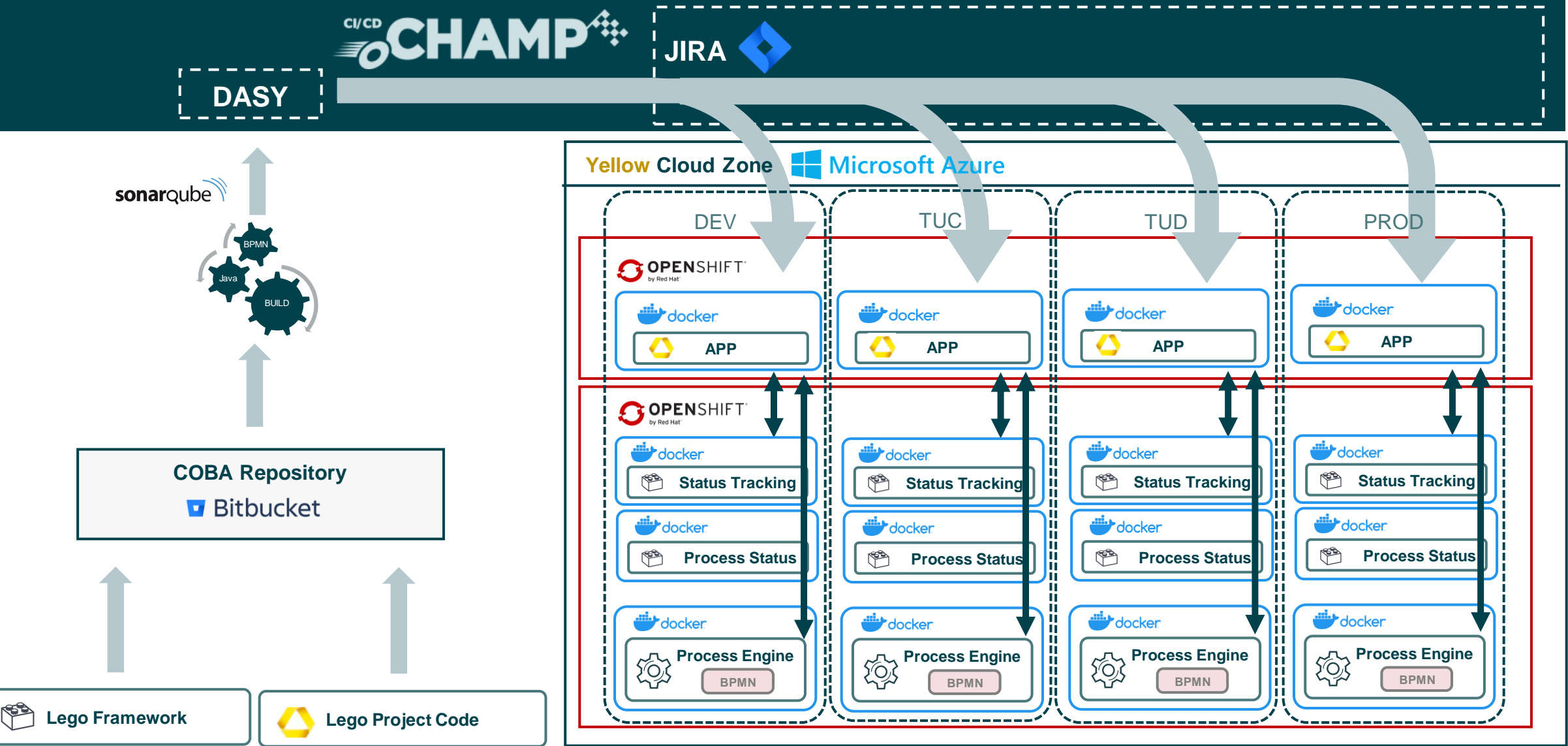
- Herstellerfunktionalität wird z.T. nachgebaut (funktionale Untermenge von Operate und Optimize)
- Instrumentierung der Process Engines notwendig (z.B. Exporter in Camunda 8)
- Ggf. zusätzliche Lizenzkosten und Resource Overhead durch zusätzliche Prozessinstanzen

Aktueller Stand



- Pilotprojekt auf dem Weg in Produktion
- Framework für Camunda 8 produktiv
- POC mit IBM durchgeführt
- Anbindung Ticketsystem in Arbeit
- Instrumentierung Prozessengines in Arbeit
- zentrale Provisionierung von Camunda 8 und IBM BAW

Infrastruktur inkl. CI /CD Integration





- Implementierung und Einbindung weiterer End2End Prozesse
- Produktivstellung Framework für IBM BAW
- Verknüpfung von Camunda Web Modeler und Process Repository



COMMERZBANK